

REMARKS

In response to the Final Office Action dated June 15, 2005, Applicant respectfully requests favorable reconsideration of the above-captioned application in view of the above-identified amendments and the following remarks. Claims 1-44 are now pending in this application.

Status of Application

To ensure entry of this Response, a Request for Continued Examination (RCE) is being filed concurrently herewith.

Overview of Changes Made to the Claims

This Response amends the preambles of claims 30-32 so that they conform to the format adopted by the other pending dependent claims. This Response also adds new dependent claims 33-44.

Regarding the 35 U.S.C. § 102 Rejection

Claims 1-32 were rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,785,21 to Immerman et al. (referred to below as "Immerman"). Applicant respectfully traverses this rejection for the following reasons.

As a main thrust, Immerman is directed to a system and method for distributing a run time model to a client for execution thereat (column 2, lines 29-31). To implement this feature, Immerman discloses a Domino Off-line Service (DOLS) 62 that provides distributed computing and remote execution of various applications (column 5, lines 3-7). In operation, the DOLS 62 provides a way for browser users (at the client site) to utilize Domino Web application offline (column 20, lines 26-29). More specifically, using a

1 browser, a user can take an application offline, make changes, and synchronize those
2 changes with the online application (column 20, lines 29-32). To enable the application
3 for offline use, a Web site developer and a Web site administrator configure and set up
4 the application (column 20, lines 46-48). Once the application is enabled, the end user
5 opens the online Web application. That is, by clicking on a control or an icon, the user
6 downloads the application to a local machine (column 20, lines 53-55). The user may
7 then interact with the offline application (column 20, lines 62-67).

8 The Office Action relies principally on columns 5 and 6 of Immerman in rejecting
9 most of the independent claims in the above-captioned application. That portion
10 describes a local run time model 90. At the outset, it is important to note that the local
11 run time model 90 refers to processing performed at a client site, not the server site. See
12 for instance, column 2, lines 55-57, which states, "FIG. 2 is a diagram illustrating the
13 objects unbundled to a local run time model in support of an API for client side execution
14 of Notes."

15 The local run time model 90 comprises a hierarchy of models including object
16 data store model 92, security model 96, indexing model 98, replication model 94, agent
17 workflow model 99, and mail model 97 (column 5, lines 31-41). The data store model 92
18 specifies the level of access that users and servers have to data elements (column 5, lines
19 53-58), the security model 96 provides a collection of log-in credentials (column 5, line
20 66), the indexing model 98 provides a search index which administrators and database
21 managers may apply to databases and files (column 6, lines 15-19), the replication model
22 94 provides a series of rules describing how to organize and synchronize databases
23 (column 6, lines 25 and 26), the agent workflow model 99 implements the execution of
24 an agent (column 6, lines 42 and 43), and the mail model 97 provides rules for
25 forwarding information from one object data store location to another (column 6, lines 50

1 and 51). In this hierarchy, “the design of a parent model is a prerequisite to the design of
2 a child model” (column 5, lines 47 and 48). Immerman provides an example of what this
3 means by stating that the “security model 96 is a prerequisite to mail model 97 in the
4 sense that mail model 97 must provide for verification of the identity of users accessing
5 mail model 97 with respect to a data object” (column 6, lines 53-57). Fig. 2 shows the
6 specific design dependencies of the above-identified models.

7 The above-described subject matter is not even remotely related to what is being
8 claimed in the present application. Consider claim 1, which is reproduced below in its
9 entirety.

10
11 1. A server system, comprising:
12 one or more computers;
13 an application executing on the computers to receive and process client requests; and
14 a constraint system to constrain operation of the application according to multiple
15 different constraints, the constraint system comprising a hierarchy of constraint layers, with each
16 constraint layer containing a set of one or more constraints that customize operation of the
17 application.

18
19 First, the claim is directed to a “server system.” The server system includes, in
20 part, “an application . . . to receive and process client requests.” The claimed “constraint
21 system” itself is an element of the “server system.” In contrast, the cited portions of
22 Immerman disclose a *“local run time model” deployed at the client site*. Thus, the
23 models described in columns 5 and 6 of Immerman pertain to client-side functionality,
24 not a constraint system employed by a server system which receives and processes client
25 requests.

1 Second, claim 1 recites, in part, “a constraint system to constrain operation of the
2 application according to multiple different constraints, the constraint system comprising a
3 hierarchy of constraint layers, with each constraint layer containing a set of one or more
4 constraints that customize operation of the application.” Immerman discloses no such
5 constraint system. In citing columns 5 and 6 of Immerman, the Patent Office is
6 apparently of the opinion that the model hierarchy shown in Fig. 2 has a bearing on the
7 claimed constraint system having a hierarchy of constraint layers. But, as summarized
8 above, that hierarchy merely illustrates the design dependencies of different components
9 of the local run time model 90; for example, the security model 96 is a prerequisite to
10 mail model 97 in the sense that the mail model 97 must provide for verification of the
11 identity of users accessing mail model 97 with respect to a data object. The models *do*
12 *not* pertain to a constraint system which *constrains or customizes the operation of an*
13 *application* as claimed. More specifically, the various models (92, 94, 96, 97, 98, 99) in
14 Fig. 2 do not constrain or customize the operation of the local run time model 90, but
15 rather the models *make up (or comprise)* the model 90 itself (see column 5, lines 31-41).
16 That is, the models are integral parts of the local run time model 90, rather than
17 constraints on some other, pre-existing and independent, application.

18 In response to this argument, the Final Office Action states, in part, that “There is
19 no limitation in the claim on the functionality of the constraint or how the constraint
20 customizes the operation of the application and therefore Immerman meets the scope of
21 the claimed limitation ‘a constraint system comprising a hierarchy of constraint layers
22 that customizes the operation of the application’” (page 11, paragraph No. 3 of the Final
23 Office Action). This position is misplaced because claim 1 *does* clearly distinguish over
24 Immerman in its present form. For instance, the organization of models shown in Fig. 2
25 of Immerman does not “customize operation of the application,” according to the plain

1 meaning of the word “customize.” According to one online dictionary, to “customize”
2 means to “change to suit better: to alter something in order to make it fit somebody’s
3 requirements better,” as in “She has customized the software to suit our needs”
4 (<http://encarta.msn.com/dictionary/customize.html>). As explained above, the
5 organization of models shown in Fig. 2 of Immerman controls the design dependency of
6 the models. It does not *change* an application; it just defines how the models are put
7 together.

8 The Final Office Action also cites a portion of the Immerman’s abstract which
9 states:

10
11 DOLS provides a layered security model that allows flexibility for controlling access to all or part
12 of an application. The highest level of security is managed through a database access control list
13 (ACL). Further refinements within the security model provide access to specific documents, and
14 their views, forms or folders, and include read access lists, write access lists, form access lists and
15 readers and authors fields.

16
17 The explanation in the October 14, 2005 Advisory Action elaborates on this argument by
18 stating, in part, that “The security model may define the ability of a user to ‘read or write
19 access’ [sic] to the model and therefore customizes the operation of the application
20 according to a security model ‘constraint’.”

21 First, it is pointed out that, in making the rejection, the Patent Office is combining
22 two different concepts together. As a first concept, the Office Action cites the
23 organization of modules shown in Fig. 2. As a second concept, the Office Action cites
24 the layered nature of the security module. Since these concepts do not address the same
25 subject matter, it is not clear how the Office Action is proposing that these concepts be

1 interpreted in aggregate. As it stands, therefore, the rejection is internally inconsistent,
2 and thus incoherent.

3 Second, even if the cited passage (in the abstract) is applied for what it describes
4 by itself, this passage fails to disclose the subject matter recited in claim 1. Immerman's
5 multi-layered security model refers to client-side functionality, not functionality deployed
6 at a "server system" as claimed. For instance, note Fig. 5 of Immerman. This figure
7 shows that the security model 80 is deployed as a client local replica, not as a component
8 of a server system that receives and processes client requests.

9 For at least the above reasons, the Applicant respectfully submits that Immerman
10 neither discloses nor suggests the subject matter recited in claim 1.

11 Claims 2-8 depend from claim 1, and are therefore allowable for at least this
12 reason. Moreover, each of these claims recites additional features which are not
13 disclosed in, or suggested by, Immerman.

14 For instance, claims 2-6 recite details of different constraint layers in the
15 constraint hierarchy. Namely, claim 2 recites that the hierarchy comprises a constraint
16 layer that contains legally mandated constraints to constrain operation of the application
17 according to legal principles. Claim 3 recites that the hierarchy comprises a constraint
18 layer that contains company-mandated constraints to constrain operation of the
19 application according to preferences of a company that operates the application. Claim 4
20 recites that the hierarchy comprises a constraint layer that contains customer constraints
21 to constrain operation of the application according to preferences of customers. Claim 5
22 recites that the hierarchy comprises a constraint layer that contains cultural constraints to
23 constrain operation of the application according to cultural aspects. And claim 6 recites
24 that the hierarchy comprises a constraint layer that contains end user constraints to
25 constrain operation of the application according to preferences of an end user.

1 To repeat, Immerman's Fig. 2 shows an object data store model 92, a security
2 model 96, an indexing model 98, a replication model 94, an agent workflow model 99,
3 and a mail model 97. These models have no bearing on the claimed constraint layers that
4 contain *legally mandated constraints, company-mandated constraints, customer*
5 *constraints, cultural constraints, and end user constraints*. Further, if, in the alternative,
6 the Office Action is interpreting the claimed constraint system as Immerman's multi-
7 layer security model, the Applicant points out that different layers of a security model
8 cannot be interpreted as the above-identified specific constraint layers.

9 In rejecting these dependent claims, the Office Action also cites column 10, line
10 30 to column 11, line 28 of Immerman, as well as column 18, line 59 to column 19, line
11 62 of Immerman. The excerpt in columns 10 and 11 refers, in part, to an ID policy
12 database 114 and an ID repository database 111. The excerpt in columns 18 and 19
13 refers, in part, to a configuration document 232 having tabbed pages for basics tab 380,
14 services tab 384, schedule tab 390, and rules tab 400. This excerpt also describes an
15 offline security policy form 410, an application page 238, web control 241, and various
16 other components 246-258. First, these topics do not have anything to do with the
17 constraint layers recited in claims 2-6, namely, constraint layers pertaining to *legally*
18 *mandated constraints, company-mandated constraints, customer constraints, cultural*
19 *constraints, and end user constraints*. For example, the words "legal" and "cultural" do
20 not appear anywhere in the Immerman document. Second, even if, assuming *arguendo*,
21 that the information disclosed in these excerpts had some relevance to the claimed
22 constraint layers (which it does not), Immerman does not describe that this information is
23 arranged in a *constraint hierarchy* (as recited in claim 1).

24 As to the recitation of specific constraint layers, the October 14, 2005 Advisory
25 Action states, in part, "In fact, claim 3 states that the constraint is defined according to

1 the preferences of a company. In comparison, the security model is defined by the
2 administrator and is therefore defined according to a company's preferences." First, the
3 security model pertains to local functionality, not a constraint layer in a "server system."
4 Second, the Office Action has not shown where Immerman states that an administrator is
5 associated with a company.

6 Dependent claim 7 recites that the constraint layers are organized within the
7 hierarchy such that a first constraint layer limits a second constraint layer but the second
8 constraint layer does not limit the first constraint layer. As described above, Immerman's
9 Fig. 2 refers to design prerequisites (that is, one component must be built for another
10 component to work properly), rather than constraints in the context of the claimed
11 invention.

12 Dependent claim 8 recites that the server system (of claim 1) further comprises a
13 constraint resolver to resolve the constraint layers so that operation of the application is
14 constrained by a sum of the constraints in the layers. Immerman discloses that the
15 different models shown in Fig. 2 comprise the local run time model 90. There is
16 absolutely no hint in Immerman that these models might conflict with each other. Hence,
17 there is likewise no suggestion that Immerman's system includes a constraint resolver to
18 cope with such conflicts. Further, if the constraint layers are being interpreted as layers
19 in a security model, there is no suggestion that these layers might conflict with each
20 other, and thus no suggestion of a constraint resolver.

21 Based on at least the above reasons, the Applicant respectfully submits that
22 Immerman neither discloses nor suggests the subject matter recited in dependent claims
23 2-8.

24 The remaining previously pending claims (9-32) recite subject matter that is
25 related to various permutations of the above-discussed subject matter (in claims 1-8).

1 Therefore these claims are allowable for reasons that are related to those given above. In
2 addition, these claims recite additional subject matter which is not disclosed or suggested
3 in Immerman.

4 Consider, for instance, independent claim 9. This claim is reproduced below in its
5 entirety:

6
7 9. A server system comprising:

8 one or more computers; and

9 a multi-layer application executing on the computers to handle client requests, the multi-
10 layer application comprising:

11 a problem-solving logic layer to process the client requests according to an associated
12 problem domain, the problem-solving logic layer containing one or more execution models to
13 perform various sets of tasks when processing the client requests, the problem-solving logic layer
14 producing replies to the client requests;

15 a presentation layer to structure the replies produced by the problem-solving logic layer
16 in a manner that makes the replies presentable on various client devices; and

17 a constraint hierarchy of multiple constraint layers, each constraint layer containing a set
18 of one or more constraints that specify how the replies should be structured to customize the
19 replies for specific sets of conditions.
20

21 The Office Action again relies, in part, on columns 5 and 6 of Immerman, but Fig.
22 2 of Immerman in no way discloses or suggests a *multi-layer application* that comprises a
23 *problem-solving logic layer* and a *presentation layer*. Furthermore, Immerman in no way
24 discloses or suggests a constraint hierarchy of multiple constraint layers, each constraint
25

1 layer containing a set of one or more constraints *that specify how the replies should be*
2 *structured to customize the replies for specific sets of conditions.*

3 The rejection of claim 9 relies in part on columns 5 and 6 of Immerman. But the
4 hierarchy described there does not include a multi-layer application that includes a
5 problem-solving logic layer and a presentation layer. Further, the hierarchy described in
6 that excerpt of Immerman is used to construct the local run time model 90, not to
7 customize replies.

8 The rejection of claim 9 also identifies column 10, line 30 to column 11, line 28
9 of Immerman. This passage, in part, reads:

10
11 ID policy database 114 is a highly secure collection of security policy documents 110. It
12 is accessed by DSAPI ID generator 108 in response to a user login request on channel 307 to
13 determine the security domain of that user and determine the correct response.

14
15 While this passage mentions determining a “correct response,” this subject matter is not
16 even remotely related to constraints that specify how replies should be structured to
17 customize the replies for specific sets of conditions. To repeat, Immerman’s main thrust
18 is to download functionality to a client device to allow the client device to work offline.
19 Thus, Immerman devotes significant discussion to this download procedure. This
20 download procedure is a preparatory procedure, enabling the client device to work
21 offline; as such, this procedure does not describe the use of a server system to customize
22 replies for client devices in the manner recited in claim 9. In other words, Immerman’s
23 “correct response” has nothing to do with the customized replies recited in claim 9.

24 As another example of the deficiency of the Immerman reference, claim 10
25 (which depends on claim 9), recites that the constraint layers can be selectively added or

1 removed from the constraint hierarchy independently of other layers in the multi-layer
2 application to produce different sets of constraints. In Immerman's Fig. 2, certain models
3 are prerequisites of others, indicating that these models *cannot be added or moved*
4 *independently of each other*.

5 As another example of the deficiency of the Immerman reference, claim 30
6 (which depends on claim 1), recites that the constraints are expressed as metadata. The
7 Office Action refers to a broad swath of columns 10 and 11 of Immerman as disclosing
8 this feature. But the entire Immerman document does not even mention the word
9 "metadata," so Immerman *nowhere* discloses the claimed subject matter.

10 For at least the above exemplary (and non-exhaustive) reasons, the Applicant
11 submits that Immerman does not anticipate any of claims 1-32, and respectfully requests
12 that this rejection be withdrawn. Namely, as stated in MPEP § 2131, a claim is
13 anticipated only if each and every element as set forth in the claim is found, either
14 expressly or inherently described, in a single prior art reference. *Verdegaal Bros. v.*
15 *Union Oil Co. of California*, 2 USPQ2d 1051 (Fed. Cir. 1987). Since Immerman does
16 not set forth each and every feature, it fails to anticipate the claims under § 102.
17 Moreover, for the reasons stated above, Immerman discloses a very different system than
18 the claimed invention, and therefore also does not render the claims obvious under 35
19 U.S.C. § 103.

20 In any event, MPEP § 707.07(f) states that, "Where the applicant traverses any
21 rejection, the examiner should, if he or she repeats the rejection, take note of the
22 applicant's arguments and answer the substance of it" (page 700-119, Rev. 2, May 2004).
23 The Final Office Action did not address the majority of the points made in the March 30,
24 2005 Response, and therefore fails to comply with the policy set forth in the MPEP. If
25 the rejection is repeated, the Patent Office is respectfully requested to address each of the

1 points raised above. A non-exhaustive list of these points is presented again here in
2 summary fashion:

3 • Some of the claims are directed to a “server system.” In contrast, the thrust in
4 Immerman is to download functionality to a client device, so that the *client device can*
5 *work offline*. This means that the server-centric role recited in some claims is nowhere
6 disclosed or suggested by the Immerman document.

7 • Some of the claims recite a constraint hierarchy that customizes an application.
8 In contrast, Immerman discloses a local run time model having component models that
9 have a prescribed design dependency. But these component models cannot be said to
10 customize an application in the manner claimed, given the plain meaning of the word
11 “customize.”

12 • Some of the claims recite constraint layers that contain legally mandated
13 constraints, company-mandated constraints, customer constraints, cultural constraints,
14 and end user constraints. Immerman discloses none of these models. The Office Action
15 has not even made clear how Immerman’s models (object data store model 92, security
16 model 96, indexing model 98, replication model 94, agent workflow model 99, and mail
17 model 97) are being construed to meet the claim language.

18 • Some of the claims recite a constraint resolver to resolve the constraint layers so
19 that operation of the application is constrained by a sum of the constraints in the layers.
20 There is absolutely no hint in Immerman that these models might conflict with each
21 other. Hence, there is likewise no suggestion that Immerman’s system includes a
22 constraint resolver to cope with such conflicts.

23 The Examiner is respectfully requested to address each of these points – and to
24 particularly address these points in a manner which does not simply involve pointing to a
25 multi-paragraph passage in Immerman. Due to the complexity and length of the

1 Immerman document, prosecution will be advanced if the Patent Office can expressly
2 name the features of Immerman that are being construed to meet the claim elements.
3 Currently, the Patent Office's correlation between the specific features recited in the
4 claims and the features of the Immerman reference is very unclear.

5
6 *New Claims*

7 This Response add new dependent claims 33-44. These claims are allowable at
8 least by virtue of their incorporation of the subject matter of the independent claims from
9 which these claims respectively depend. In addition, claims 33-44 add new features
10 which further distinguish the claimed invention over Immerman.

11 For example, claim 33 recites that "each constraint layer represents a different
12 source entity that customizes the application." The different models shown in Fig. 2 of
13 Immerman do not represent different source entities that customize an application.
14 Likewise, different layers in a security model do not represent different source entities
15 that customize an application.

16 As another example, claim 39 recites, in summary, that the hierarchy comprises
17 each of: a constraint layer that contains legally mandated constraints, a constraint layer
18 that contains company-mandated constraints, a constraint layer that contains customer
19 constraints, a constraint layer that contains cultural constraints, and a constraint layer that
20 contains end user constraints. Since Immerman does not disclose *any one* of these layers
21 individually, it certainly does not disclose the *combination* of these different layers.

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

2
3
4
5

6
7
8
9

14

15